Case Study One – Black’s Property Management

**There may be errors or ambiguities in this information – please seek clarification if necessary. Also, please note that this information may change for your benefit.**

Black’s Property Management is a property management company based in Auckland. The director, Tane Black, wants to keep track of landlords, properties, jobs, tradesmen, and materials. The current informal paper-based system often causes problems with invoicing landlord’s jobs and with keeping records of tradesmen and the materials used.

Tane Black needs the application to be written in Visual C# and be able to run on Microsoft Visual Studio 2019 using an MS Access database that will be provided to you. You are to use the **OleDb** objects (set up in the data controller) to interact with the MS Access database tables. Tane has also requested that you are to use programming logic (i.e., master-detail relationships in conjunction with ‘for-each’ loops and find in conjunction with data views) to create the reports. Tane realises that this setup might but not be using the most up-to-date technology, but he knows that this technology is sufficient to meet his requirements without incurring extra costs and without any changes to his system.

Use Case Diagram



1. Display Maintenance Menu – the screen that the maintenance clerk requires to start the application and select options.

|  |  |  |
| --- | --- | --- |
| **USE CASE NAME:** | Display Maintenance Menu | **USE CASE TYPE** |
| **USE CASE ID:** | 1 | **Design Requirements: þ** |
| **PRIORITY:** | High |  |
| **PRIMARY BUSINESS ACTOR:** | Maintenance Clerk | |
| **DESCRIPTION:** | This use case describes the maintenance clerk starting the application and selecting options. | |
| **PRE-CONDITIONS:** | The maintenance clerk has logged onto the system | |
| **TYPICAL COURSE** | Step 1: The maintenance clerk starts the “Black’s Property Management” application.  Step 2: The system creates the data controller object that connects the application to the database.  Step 3: The system displays the “Maintenance Menu” form with the following options:  Add a property.  Update a property.  Delete a property.  Add a job.  Update a job.  Delete a job.  Assign material.  Remove material.  Print invoices.  Print tradesmen report.  Step 4: The maintenance clerk selects an option.  Step 5: The system loads the corresponding form.  Step 6: The maintenance clerk clicks on the “Exit” button.  Step 7: The system closes the application. | |
| **OF EVENTS:** |
|  |
|  |
|  |
|  |
| **ALTERNATE COURSES:** | Step 4a1: The maintenance clerk clicks on the “Exit” button.  Step 4a2: The system goes to step 7. | |
| Step 6a1: The maintenance clerk elects to select another option.  Step 6a2: The system goes back to step 3. | |
| **POST CONDITIONS:** | None | |
| **ASSUMPTIONS:** | None | |

1. Add Property – the screen that the maintenance clerk requires to add a property’s details.

|  |  |  |
| --- | --- | --- |
| **USE CASE NAME:** | Add Property | **USE CASE TYPE** |
| **USE CASE ID:** | 2 | **Design Requirements þ** |
| **PRIORITY:** | High |  |
| **PRIMARY BUSINESS ACTOR:** | Maintenance Clerk | |
| **DESCRIPTION:** | This use case describes the maintenance clerk adding a property’s details to the system. | |
| **PRE-CONDITIONS:** | The maintenance clerk has logged on to the system | |
| **TYPICAL COURSE**  **OF EVENTS:** | Step 1 The maintenance clerk selects the “Add Property” function.  Step 2 The system displays the “Add Property” form.  Step 3 The maintenance clerk enters the property’s street address, suburb, status, and year built.  Step 4 The system gets all landlords.  Step 5 The system displays the list of landlords in combo boxes: landlord id, landlord last name, and landlord first name.  Step 6 The maintenance clerk selects a landlord.  Step 7 The system gets the details of the selected landlord.  Step 8 The system displays the selected landlord’s details in read-only text boxes: landlord last name, landlord first name, street address, and suburb.  Step 9 The maintenance clerk clicks on the “Add Property” button.  Step 10 The system confirms that all necessary fields are filled in correctly.  Step 11 The system gets a unique property id.  Step 12 The system saves the property’s details: property id, street address, suburb, status, year built, and landlord id.  Step 13 The system displays the “Property added successfully” message.  Step 14 The maintenance clerk clicks on “OK” in the message box.  Step 15 The system resets the street address, suburb, status, year-built, landlord last name, landlord first name, street address, and suburb.fields.  Step 16 The maintenance clerk clicks on the “Return” button.  Step 17 The system closes the form to end the use case. | |
| **ALTERNATE COURSES:** | Step 9a.1 The maintenance clerk clicks on the “Return” button.  Step 9a.2The system goes to step 17. | |
| Step 11a.1 The system, having identified incorrect fields, displays the “Please fill in all fields correctly” message.  Step 11a.2 The maintenance clerk clicks on “OK” in the message box.  Step 11a.3 The system goes to step 3. | |
| Step 17a.1 The maintenance clerk elects to add another property.  Step 17a.2 The system goes to step 3. | |
| **POST CONDITIONS:** | None. | |
| **ASSUMPTIONS:** | None. | |

1. Update Property – the screen that the maintenance clerk requires to update a property’s details.

|  |  |  |
| --- | --- | --- |
| **USE CASE NAME:** | Update Property | **USE CASE TYPE** |
| **USE CASE ID:** | 3 | **Design Requirements þ** |
| **PRIORITY:** | High |  |
| **PRIMARY BUSINESS ACTOR:** | Maintenance Clerk | |
| **DESCRIPTION:** | This use case describes the maintenance clerk updating an existing property’s details. | |
| **PRE-CONDITIONS:** | The maintenance clerk has logged on to the system. | |
| **TYPICAL COURSE**  **OF EVENTS:** | Step 1 The maintenance clerk selects the “Update Property” function.  Step 2 The system displays the “Update Property” form.  Step 3 The system gets all properties.  Step 4 The system displays the list of all properties in a data grid view: property id, street address, suburb, status, year built, and landlord id.  Step 5 The maintenance clerk selects a property.  Step 6 The system gets the selected property’s details.  Step 7 The system displays the property’s details: property ID, street address, suburb, status, year built, landlord last name, and landlord first name.  Step 8 The maintenance clerk modifies the relevant details: street address, suburb, status, and year built only.  Step 9 The maintenance clerk clicks on the “Update Property” button.  Step 10 The system confirms that all necessary fields are filled in correctly.  Step 11 The system updates the property’s details: street address, suburb, status, and year built.  Step 12 The system displays the “Property updated successfully” message.  Step 13 The maintenance clerk clicks on “OK” in the message box.  Step 14 The maintenance clerk clicks on the “Return” button.  Step 15 The system closes the form to end the use case. | |
|  |
|  |
|  |
|  |
| **ALTERNATE COURSES:** | Step 9a.1 The maintenance clerk clicks on the “Return” button.  Step 9a.2 The system goes to step 15. | |
| Step 11a.1 The system, having identified incorrect fields, displays the “Please fill in all fields correctly” message.  Step 11a.2 The maintenance clerk clicks on “OK” in the message box.  Step 11a.3 The system goes to step 8. | |
| Step 14a.1 The maintenance clerk elects to update another property.  Step 14a.2 The system goes to step 3. | |
| **POST CONDITIONS:** | None. | |
| **ASSUMPTIONS:** | None. | |

1. Delete Property – the screen that the maintenance clerk requires to delete a property.

|  |  |  |
| --- | --- | --- |
| **USE CASE NAME:** | Delete Property | **USE CASE TYPE** |
| **USE CASE ID:** | 4 | **Design Requirements þ** |
| **PRIORITY:** | High |  |
| **PRIMARY BUSINESS ACTOR:** | Maintenance Clerk | |
| **DESCRIPTION:** | This use case describes the maintenance clerk deleting the details of an existing property that is no longer required. | |
| **PRE-CONDITIONS:** | The maintenance clerk has logged on to the system. | |
| **TYPICAL COURSE** | Step 1 The maintenance clerk selects the “Delete Property” function.  Step 2 The system displays the “Delete Property” form.  Step 3 The system gets all properties.  Step 4 The system displays the list of properties in a data grid view: property id, street address, suburb, status, year built, and landlord id.  Step 5 The maintenance clerk selects a property.  Step 6 The system gets the selected property’s details.  Step 7 The system displays the property’s details in read-only text boxes: property id, street address, year built, landlord last name, and landlord first name.  Step 8 The maintenance clerk clicks on the “Delete Property” button.  Step 9 The system confirms that the property is not associated with any jobs.  Step 10 The system displays the “Are you sure you want to delete this property?” message.  Step 11The maintenance clerk clicks on “OK” in the message box  Step 12 The system deletes the property’s details.  Step 13 The system displays the “Property deleted successfully” message.  Step 14 The maintenance clerk clicks on “OK” in the message box.  Step 15 The system clears the property id, street address, year built, landlord last name, and landlord first name fields.  Step 16 The maintenance clerk clicks on the “Return” button.  Step 17 The system closes the form to end the use case. | |
| **OF EVENTS:** |
|  |
|  |
|  |
|  |
| **ALTERNATE COURSES:** | Step 8a.1 The maintenance clerk clicks on the “Return” button.  Step 8a.2 The system goes to step 15. | |
| Step 10a.1 The system displays the message “Property’s with jobs cannot be deleted.”  Step 10a.2 The maintenance clerk clicks on “OK” in the message box.  Step 10a.3 The system goes to step 3. | |
| Step 11a.1 The maintenance clerk clicks on “Cancel” in the message box.  Step 11a.2 The system goes to step 5. | |
| Step 14a.1 The maintenance clerk elects to delete another property.  Step 14a.2 The system goes to step 3. | |
| **POST CONDITIONS:** | None. | |
| **ASSUMPTIONS:** | None. | |

1. Add Job – the screen that the maintenance clerk requires to add a job.

|  |  |  |
| --- | --- | --- |
| **USE CASE NAME:** | Add Job | **USE CASE TYPE** |
| **USE CASE ID:** | 5 | **Design Requirements þ** |
| **PRIORITY:** | High |  |
| **PRIMARY BUSINESS ACTOR:** | Maintenance Clerk | |
| **DESCRIPTION:** | This use case describes the maintenance clerk adding a job’s details to the system. | |
| **PRE-CONDITIONS:** | The maintenance clerk has logged on to the system | |
| **TYPICAL COURSE** | Step 1 The maintenance clerk selects the “Add Job” function.  Step 2 The system displays the “Add Job” form.  Step 3 The system gets all landlords with a valid credit status.  Step 4 The system displays the list of landlords with a valid credit status in a list box: landlord id, landlord last name, and landlord first name.  Step 5 The maintenance clerk selects a landlord.  Step 6 The system gets all properties owned by the selected landlord.  Step 7 The system displays the list of properties owned by the selected landlord in a list box: property id, street address, suburb, and status.  Step 8 The maintenance clerk selects a property.  Step 9 The system gets all tradesmen.  Step 10 The system displays the list of tradesmen in combo boxes: tradesman id, tradesman last name, and tradesman first name.  Step 11 The maintenance clerk selects a tradesman.  Step 12 The maintenance clerk enters the job date, job description, and job fee.  Step 13 The maintenance clerk clicks on the “Add Job” button.  Step 14 The system confirms that all necessary fields are filled in correctly.  Step 15 The system gets a unique job id for the new job.  Step 16 The system saves the job’s details: job id, job description, job fee, job date, job status: set to current, property id, and tradesman id.  Step 17 The system displays the “Job added successfully” message.  Step 18 The maintenance clerk clicks on “OK” in the message box.  Step 19 The system rests the job date, job description, and job fee fields.  Step 20 The maintenance clerk clicks on the “Return” button.  Step 21 The system closes the form to end the use case. | |
| **OF EVENTS:** |
|  |
|  |
|  |
|  |
| **ALTERNATE COURSES:** | Step 13a.1 The maintenance clerk clicks on the “Return” button.  Step 13a.2 The system goes to step 21. | |
| Step 15a.1 The system, having identified incorrect fields, displays the “Please fill in all fields correctly” message.  Step 15a.2 The maintenance clerk clicks on “OK” in the message box.  Step 15a.3 The system goes to step 12. | |
| Step20a.1 The maintenance clerk elects to add another job.  Step 20a.2 The system goes to step 5. | |
| **POST CONDITIONS:** | None. | |
| **ASSUMPTIONS:** | None. | |

1. Update Job – the screen that the maintenance clerk requires to update a job’s details.

|  |  |  |
| --- | --- | --- |
| **USE CASE NAME:** | Update Job | **USE CASE TYPE** |
| **USE CASE ID:** | 6 | **Design Requirements þ** |
| **PRIORITY:** | High |  |
| **PRIMARY BUSINESS ACTOR:** | Maintenance Clerk | |
| **DESCRIPTION:** | This use case describes the maintenance clerk updating the details of an existing job. | |
| **PRE-CONDITIONS:** | The maintenance clerk has logged on to the system. | |
| **TYPICAL COURSE** | Step 1 The maintenance clerk selects the “Update Job” function.  Step 2 The system displays the “Update Job” form.  Step 3 The system gets all jobs.  Step 4 The system displays the list of jobs in a data grid view: job id, job description, job status, property id, and landlord id.  Step 5 The maintenance clerk selects a job.  Step 6 The system get the selected job’s details.  Step 7 The system displays the job’s details: job id, job description, job date, job fee, job status, property id, property street address, landlord id, landlord last name, and landlord first name.  Step 8 The maintenance clerk modifies the relevant details: job description, job date, job fee, and job status only.  Step 9 The maintenance clerk clicks on the “Update Job” button.  Step 10 The system confirms that all necessary fields are filled in correctly.  Step 11 The system confirms that the job status is current.  Step 12 The system updates the job’s details: job description, job date, job fee, and job status.  Step 13 The system displays the “Job updated successfully” message.  Step 14 The maintenance clerk clicks on “OK” in the message box.  Step 15 The maintenance clerk clicks on the “Return” button.  Step 16 The system closes the form to end the use case. | |
| **OF EVENTS:** |
|  |
|  |
|  |
|  |
| **ALTERNATE COURSES:** | Step 9a.1 The maintenance clerk clicks on the “Return” button.  Step 9a.2 The system goes to step 16. | |
| Step 11a.1 The system displays the “Paid jobs cannot be updated” message.  Step 11a.2 The maintenance clerk clicks on “OK” in the message box.  Step 11a.3 The system goes to step 3. | |
| Step 12a.1 The system, having identified incorrect fields, displays the “Please fill in all fields correctly” message.  Step 12a.2 The maintenance clerk clicks on “OK” in the message box.  Step 12a.3 The system goes to step 8. | |
| Step 15a.1 The maintenance clerk elects to update another job.  Step 15a.2 The system goes to step 3. | |
| **POST CONDITIONS:** | None. | |
| **ASSUMPTIONS:** | None. | |

1. Delete Job – the screen that the maintenance clerk requires to delete a job.

|  |  |  |
| --- | --- | --- |
| **USE CASE NAME:** | Delete Job | **USE CASE TYPE** |
| **USE CASE ID:** | 7 | **Design Requirements þ** |
| **PRIORITY:** | High |  |
| **PRIMARY BUSINESS ACTOR:** | Maintenance Clerk | |
| **DESCRIPTION:** | This use case describes the maintenance clerk deleting the details of an existing job that is no longer required. | |
| **PRE-CONDITIONS:** | The maintenance clerk has logged on to the system. | |
| **TYPICAL COURSE**  **OF EVENTS:** | Step 1 The maintenance clerk selects the “Delete Job” function.  Step 2 The system displays the “Delete Job” form.  Step 3 The system gets all jobs.  Step 4 The system displays the list of all jobs in a data grid view: job id, job description, job date, job status, job fee, property id, and tradesman id  Step 5 The maintenance clerk selects a job.  Step 6 The system gets the selected job’s details.  Step 7 The system displays the job’s details in read-only text boxes: job id, job description, job fee, job date, and job status.  Step 8 The maintenance clerk clicks on the “Delete Job” button.  Step 9 The system confirms that the job has been paid.  Step 10 The system displays the “Are you sure you want to delete this job?” prompt.  Step 11 The maintenance clerk clicks on “OK” in the message box.  Step 12 The system deletes all job materials related to the job.  Step 13 The system deletes the job.  Step 14 The system displays the “Job deleted successfully” message.  Step 15 The maintenance clerk clicks on “OK” in the message box.  Step 16 The system clears the job id, job description, job fee, job date, and job status fields.  Step 17 The maintenance clerk clicks on the “Return” button.  Step 18 The system closes the form to end the use case. | |
| **ALTERNATE COURSES:** | Step 8a.1 The maintenance clerk clicks on the “Return” button.  Step 8a.2 The system goes to step 18. | |
| Step 10a.1 The system displays the message “Current jobs cannot be deleted.”  Step 10a.2 The maintenance clerk clicks on “OK” in the message box.  Step 10a.3 The system goes to step 3. | |
| Step 11a.1 The maintenance clerk clicks on “Cancel” in the message box.  Step 11a.3 The system goes to step 3. | |
| Step 17a.1 The maintenance clerk elects to delete another job.  Step 17a.2 The system goes to step 3. | |
| **POST CONDITIONS:** | None. | |
| **ASSUMPTIONS:** | None. | |

1. Assign Material – the screen that the maintenance clerk requires to assign a material to a job.

|  |  |  |
| --- | --- | --- |
| **USE CASE NAME:** | Assign Material | **USE CASE TYPE** |
| **USE CASE ID:** | 8 | **Design Requirements: þ** |
| **PRIORITY:** | High |  |
| **PRIMARY BUSINESS ACTOR:** | Maintenance Clerk | |
| **DESCRIPTION:** | This use case describes the maintenance clerk assigning a material to a job. | |
| **PRE-CONDITIONS:** | The maintenance clerk has logged on to the system. | |
| **TYPICAL COURSE** | Step 1 The maintenance clerk selects the “Assign Material” function.  Step 2 The system displays the “Assign Material” form.  Step 3 The system gets all jobs.  Step 4 The system displays the list of jobs in a data grid view: job id, job description, job date, job status, job fee , landlord id, and property id.  Step 5 The maintenance clerk selects a job.  Step 6 The system gets the materials already assigned to the job.  Step 7 The system displays the materials already assigned to the job: job id, material id and quantity.  Step 8 The system gets all materials.  Step 9 The system displays the list of materials in a data grid view: material id, material description, and cost.  Step 10 The maintenance clerk selects a material.  Step 11 The maintenance clerk enters the quantity assigned using a number picker.  Step 12 The maintenance clerk clicks on the “Assign Material” button.  Step 13 The system confirms that the selected job is current.  Step 14 The system saves the job material details: job id, material id, and quantity.  Step 15 The system displays the “Material assigned successfully” message.  Step 16 The maintenance clerk clicks on “OK” in the message”.  Step 17 The maintenance clerk clicks on the “Return” button.  Step 18 The system closes the form to end the use case. | |
| **OF EVENTS:** |
|  |
|  |
|  |
|  |
| **ALTERNATE COURSES:** | Step 12a.1 The maintenance clerk clicks on the “Return” button.  Step 12a.2 The system goes to step 18. | |
| Step 14a.1 The system displays the “Cannot assign material to a paid job” message.  Step 14a.2 The maintenance clicks on “OK” in the message box.  Step 14a.3 The system goes to step 4. | |
| Step 15a.1 The system displays the error message “This material is already assigned to this job.”  Step 15a.2 The maintenance clerk clicks on “OK” in the message”.  Step 15a.3 The system goes to step 8. | |
| Step 17a.1 The maintenance clerk elects to assign another material.  Step 17a.2 The system goes to step 3. | |
| **POST CONDITIONS:** | None | |
| **ASSUMPTIONS:** | None | |

1. Remove Material – the screen that the maintenance clerk requires to remove material from a job.

|  |  |  |
| --- | --- | --- |
| **USE CASE NAME:** | Remove Material | **USE CASE TYPE** |
| **USE CASE ID:** | 9 | **Design Requirements: þ** |
| **PRIORITY:** | High |  |
| **PRIMARY BUSINESS ACTOR:** | Maintenance Clerk | |
| **DESCRIPTION:** | This use case describes the maintenance clerk removing a material from a job. | |
| **PRE-CONDITIONS:** | The maintenance clerk has logged on to the system. | |
| **TYPICAL COURSE**  **OF EVENTS:** | Step 1 The maintenance clerk selects the “Remove Material” function.  Step 2 The system displays the “Remove Material” form.  Step 3 The system gets all jobs.  Step 4 The system displays the list of jobs: job id, job description, job date, and job status.  Step 5 The maintenance clerk selects a job.  Step 6 The system gets the materials assigned to the job.  Step 7 The system displays the materials assigned to the job: job id, material id, and quantity.  Step 8 The maintenance clerk selects a material.  Step 9 The gets the material details.  Step 10 The system displays the materials details in read-only text boxes: material description and cost.  Step 11 The maintenance clerk clicks on the “Remove Material” button.  Step 12 The system confirms that the status of the job is current.  Step 13 The system deletes the job material details.  Step 14 The system displays the “Material removed successfully” message.  Step 15 The maintenance clerk clicks on “OK” in the message” box.  Step 16 The system clears the material description and cost fields.  Step 17 The maintenance clerk clicks on the “Return” button.  Step 18 The system closes the form to end the use case. | |
| **ALTERNATE COURSES:** | Step 11a.1 The maintenance clerk clicks on the “Return” button.  Step 11a.2 The system goes to step 18. | |
| Step 13a.1 The system displays the “Cannot remove material from a paid job” message.  Step 13a.2 The maintenance clerk clicks on “OK” in the message box.  Step 13a.3 The system goes to step 5. | |
| Step 17a.1 The maintenance clerk elects to remove another material.  Step 17a.2 The system goes to step 3. | |
| **POST CONDITIONS:** | None | |
| **ASSUMPTIONS:** | None | |

1. Print Tradesmen Report – the screen that the maintenance clerk requires to print a report of the tradesmen.

|  |  |  |
| --- | --- | --- |
| **USE CASE NAME:** | Print Tradesmen Report | **USE CASE TYPE** |
| **USE CASE ID:** | 10 | **Design Requirements þ** |
| **PRIORITY:** | High |  |
| **PRIMARY BUSINESS ACTOR:** | Maintenance Clerk | |
| **DESCRIPTION:** | This use case describes the maintenance clerk printing the tradesmen report. | |
| **PRE-CONDITIONS:** | The maintenance clerk has logged on to the system. | |
| **TYPICAL COURSE**  **OF EVENTS:** | Step 1 The maintenance clerk selects the “Print Tradesmen” function.  Step 2 The system displays the “Print Tradesmen” form.  Step 3 The maintenance clerk clicks on the “Print Tradesmen” button to generate the report.  Step 4 The system gets the details of each tradesman: tradesman id, tradesman last name, tradesman first name, phone number, and fee.  Step 4.1 The system gets the details of each of the tradesman’s jobs: job id, job description, job date, job fee, and job status.  Step 4.1.1 The system gets the details of the job’s property: property id, street address, suburb, and year built.  Step 4.2 The system gets the material id and quantity of each of the job’s materials.  Step 4.2.1 The system gets the material description and cost of each’s material.  Step 5 The system prints the tradesmen report: tradesman id, tradesman last name, tradesman first name, phone number, fee, job id, job description, job fee, job date, job status, property id, street address, suburb, year built, material id, material description, cost, and quantity for each tradesman with one tradesman per page - see sample report 1.  Step 6 The maintenance clerk clicks on the “Return” button.  Step 7 The system closes the form to end the use case. | |
| **ALTERNATE COURSES:** | Step 3a.1 The maintenance clerk clicks on the “Return” button.  Step 3a.2 The system goes to step 7. | |
| Step 4.1a.1 The system prints “This tradesman has no jobs currently” on the report for this tradesman | |
| Step 4.2a.1 The system prints “There are no materials for this job” on the report for this job. | |
| **POST CONDITIONS:** | None. | |
| **ASSUMPTIONS:** | None. | |

Sample Report 1: Tradesmen Report

**Please note that this is a rough draft, and you may fine-tune the layout if you wish.**

**Black’s Property Management**

**Tradesmen Report**

**Tradesman ID: 2**

**Name: Page, Robert Fee: $50.00**

**Phone number: 0276633229**

**Job ID: 2 Date: 30/1/2023 Job Status: Current**

**Description: Fence mending Fee: $60.00**

**Property ID: 5 Address: 4 Jade Lane, Avondale**

**Year Built: 1960**

**Materials: Description Cost Quantity**

**ID:1 Fence Post $40.00 1**

**ID:2 10 Inch Nails $0.20 20**

**ID:3 1 Metre Fence Wire $15.00 2**

**Job ID: 3 Date: 31/1/2023 Job Status: Current**

**Description: Shower blocked Fee: $50.00**

**Property ID: 1 Address: 56 Kings Road, New Lynn**

**Year Built: 1950**

**Materials: Description Cost Quantity**

**ID:4 Drain Plunger $12.00 1**

**ID:5 Drain Cleaner Fluid $8.50 1**

1. Print Invoices – the screen that the maintenance clerk requires to print the invoices for all current jobs.

|  |  |  |
| --- | --- | --- |
| **USE CASE NAME:** | Print Invoices | **USE CASE TYPE** |
| **USE CASE ID:** | 11 | **Design Requirements þ** |
| **PRIORITY:** | High |  |
| **PRIMARY BUSINESS ACTOR:** | Maintenance Clerk | |
| **DESCRIPTION:** | This use case describes the maintenance clerk printing the invoices for all current jobs. | |
| **PRE-CONDITIONS:** | The maintenance clerk has logged on to the system. | |
| **TYPICAL COURSE**  **OF EVENTS:** | Step 1 The maintenance clerk selects the “Print Invoices” function.  Step 2 The system displays the “Print Invoices” form.  Step 3 The maintenance clerk clicks on the “Print Invoices” button to generate the invoices.  Step 4 The system gets the details of each current job: job id, job description, job fee, and job date.  Step 4.1 The system get the details of the job’s property: property id, street address, suburb, and landlord id.  Step 4.2 The system get the details of the property’s landlord: landlord last name, landlord first name, street address, and suburb.  Step 4.3 The system get the details of the job’s tradesman: tradesman id, tradesman last name, tradesman first name, and fee.  Step 4.4 The system gets the material id and quantity of each of the job’s materials.  Step 4.4.1 The system gets the material description and cost of each material.  Step 4.4.2 The system calculates the material price by multiplying each quantity by the material cost.  Step 4.5 The system calculates the material total by summing the material prices for each job.  Step 4.6 The system calculates each job’s total by adding the job fee, the tradesman’s fee, and the material total together.  Step 5 The system prints the invoices: landlord id, landlord last name, landlord first name, street address, suburb, property id, street address, suburb, job id, job description, job fee, job date, tradesman id, tradesman last name, tradesman first name, fee, material id, material description, cost, quantity, material price, material total, and job total for each job with one invoice per page - see sample report 2.  Step 6 The maintenance clerk clicks on the “Return” button.  Step 7 The system closes the form to end the use case. | |
|  |
|  |
|  |
|  |
| **ALTERNATE COURSES:** | Step 3a.1 The maintenance clerk clicks on the “Return” button.  Step 3a.2 The system goes to step 7. | |
| Step 4.4a.1 The system prints “There are no materials for this job” on the report for this job. | |
| **POST CONDITIONS:** | None. | |
| **ASSUMPTIONS:** | None. | |

Sample Report 2: Invoices

**Please note that this is a rough draft, and you may fine-tune the layout if you wish.**

**Black’s Property Management**

**Invoice**

**Ian King**

**21 Kings Avenue**

**Blockhouse Bay**

**Landlord ID: 1**

**Job ID: 2 Date: 30/1/2023**

**Property ID: 5 Address: 4 Jade Lane, Avondale**

**Tradesman: Robert Page $50.00**

**Description: Fence mending $60.00**

**Materials:**

**ID Description Cost Quantity Price**

**1 Fence Post $40.00 1 $40.00**

**2 10 Inch Nails $0.20 20 $4.00**

**3 1 Metre Fence Wire $15.00 2 $30.00**

**Total Materials: $74.00**

**Job Total: $184.00**

Overall Class Diagram



Logical Database Design: ERD



##### Data Dictionary

**Please remember that Tane has spoken to his staff, and they all agree that they require the size of text boxes to reflect the maximum size of the corresponding data.**

**LANDLORD**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field Name** | **Required** | **Data Type** | **Maximum Length** | **Range/List/Format** |
| **LandlordID** | **Yes** | **Auto-number** | **5** | **Primary Key** |
| **LandlordLastName** | **Yes** | **Text** | **25** | **-** |
| **LandlordFirstName** | **Yes** | **Text** | **25** | **-** |
| **StreetAddress** | **Yes** | **Text** | **50** | **-** |
| **Suburb** | **Yes** | **Text** | **20** | **-** |
| **PhoneNumber** | **No** | **Text** | **16** | **-** |
| **CreditStatus** | **Yes** | **Text** | **7** | **Valid, Invalid** |

**JOB**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field Name** | **Required** | **Data Type** | **Maximum Length** | **Range/List/Format** |
| **JobID** | **Yes** | **Auto-number** | **8** | **Primary Key** |
| **JobDescription** | **Yes** | **Text** | **100** | **-** |
| **JobDate** | **Yes** | **Date** | **10** | **DD/MM/YYYY** |
| **JobStatus** | **Yes** | **Text** | **7** | **Current, Paid** |
| **JobFee** | **Yes** | **Currency** | **6** | **50.00 – 150.00 inclusive** |
| **PropertyID** | **Yes** | **Foreign Key (PROPERTY)** | **5** | **-** |
| **TradesmanID** | **Yes** | **Foreign Key (TRADESMAN)** | **3** | **-** |

**PROPERTY**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field Name** | **Required** | **Data Type** | **Maximum Length** | **Range/List/Format** |
| **PropertyID** | **Yes** | **Auto-number** | **5** | **Primary Key** |
| **StreetAddress** | **Yes** | **Text** | **60** | **-** |
| **Suburb** | **Yes** | **Text** | **20** | **-** |
| **Status** | **Yes** | **Text** | **10** | **Occupied, Unoccupied** |
| **YearBuilt** | **Yes** | **Number** | **4** | **1900 to 2022 inclusive** |
| **LandlordID** | **Yes** | **Foreign Key (LANDLORD)** | **5** | **-** |

**TRADESMAN**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field Name** | **Required** | **Data Type** | **Maximum Length** | **Range/List/Format** |
| **TradesmanID** | **Yes** | **Auto Number** | **3** | **Primary Key** |
| **TradesmanLastName** | **Yes** | **Text** | **30** | **-** |
| **TradesmanFirstName** | **Yes** | **Text** | **30** | **-** |
| **PhoneNumber** | **Yes** | **Text** | **16** | **-** |
| **Fee** | **Yes** | **Currency** | **5** | **40.00 – 120.00 inclusive** |

**MATERIAL**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field Name** | **Required** | **Data Type** | **Maximum Length** | **Range/List/Format** |
| **MaterialID** | **Yes** | **Auto Number** | **7** | **Primary Key** |
| **MaterialDescription** | **Yes** | **Text** | **50** | **-** |
| **Cost** | **Yes** | **Currency** | **7** | **0.10 – 4000.00 inclusive** |

**JOBMATERIAL**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field Name** | **Required** | **Data Type** | **Maximum Length** | **Range/List/Format** |
| **JobID** | **Yes** | **Foreign Key (JOB)** | **8** | **Primary Key** |
| **MaterialID** | **Yes** | **Foreign Key (MATERIAL)** | **7** |
| **Quantity** | **Yes** | **Integer** | **2** | **1-40 inclusive** |